

Computational Modeling & Simulation I (Interdisciplinary Science)
CS 346

Spring 2021

Homework 1, Stage 2

Due **BY 11:59PM** Wednesday, March 24

(Please see the notes below!)

- This is a **team assignment, with restricted collaboration**: You will be working in teams of two, and you may not consult with other students (whether or not they're in CS346) or online sources (other than those on the course website or from mathworks.com) as part of the exercise.
- Teams have been selected (randomly, with a few modifications) and posted on the HW page of the course website.
- A little reminder: As part of naming conventions for the course, please make sure an identifier such as your name or userid is part of every file you submit. For exercises in CS346 done in teams of two, each filename should contain the names / userids of both people on the team.
- Each team should submit a single `.m` file and a single PDF document for this assignment. Please use the string `HW1S2` (for “HW 1, Stage 2”) in the folder name for submitting this assignment; some variant of that (e.g., `hw1s2`) could be a useful part of filenames for this assignment. See documents on the course website for additional HW guidelines and submission instructions.

HW1, Stage 2: Reflect and Revise!

Please recall that this is a **collaboration-restricted** assignment: You should work with your teammate but not discuss the assignment with others and not use online sources for assistance (except as noted above).

For HW1, Stage 2, you and a teammate should work together to do the following:

1. Read over the code that each of you submitted for HW1, exercise 3, the “1-D random walkers” exercise (“**A Staggering Exercise?**”). (To be clear: In this exceptional instance, looking at each other’s code is *required*.) Working together, analyze each code submission: For each body of code, you as a team should identify and list things that you found beneficial and things that could be improved. To guide you, please consider not just whether or not the code gives correct output, but also all of the criteria relevant for good code in this domain—e.g., easy to read and understand, easy to use, easy to analyze, easy to maintain or extend—that we have already discussed this semester.

Be sure to consider not just your own perspectives on your team’s code, but also those of potential outside readers. For example, what did you find difficult to understand in

your teammate’s code? What do you think that other intermediate-level programmers, not in Colby’s CS346, would find difficult to understand, both in your code and in your teammate’s code? Please be specific in documenting the elements of the original code submissions to which you refer. (Feel free to use line numbers from the submitted files for HW1 as part of referring to them, and if you think it improves the clarity of your report, feel free to include code with line numbers in your write-up.)

Please be thorough in your analysis and discussion with your teammate, and complete in your write-up! An analysis or report that misses important elements of correctness or style—whether from your own perspective as CS346 students or from the “outsider” perspective described above—may not receive full credit.

2. Together, as a team, create a revised version of code for HW1 ex. 3, reflecting your team’s analysis of your original submissions, both the strengths of those original submissions and the ways you identified that they could be improved. Submit the code in an aptly-named `.m` file to be easily run and evaluated.

What changes did you make in your revised code to make things easier to understand, for each other and for outside readers, and how do you believe those changes helped to make your code easier to read, understand, analyze, maintain, and extend? In your write-up, be sure to document the code thoroughly (as usual), including changes from your original versions and the reasons for those changes. Once again, please be specific in documenting the elements of code to which you refer.

3. In your write-up, please reflect briefly on this process: As you reviewed each other’s code and discussed style and technique, what did each of you learn about your own programming style by looking at and discussing each other’s code, and writing a new version reflecting your shared perspectives? If neither teammate learned anything important in the process, please say so, but be sure you have described the elements of programming style and technique you discussed as part of the process of analysis and reflection. (This does not need to be lengthy—one or two paragraphs could be more than sufficient—although please feel free to submit longer answers if you have more to say for this part of the exercise!)